

Máster Avanzado de Programación en Python para Hacking, Big Data y Machine Learning

Buenas prácticas de programación en Python

LECCIÓN 5

Control de versiones git y GitHub

INTRODUCCIÓN

En esta lección aprenderemos en qué consiste los Sistemas de Control de Versiones y por qué su utilidad lo ha convertido en una herramienta fundamental e imprescindible para todo desarrollador de software.

Adicionalmente, estudiaremos el Sistema de Control de Versiones Git y Github y cómo integrar este último en el entorno de desarrollo Visual Studio Code, así como las herramientas principales para poder crear, clonar, actualizar y descargar contenido de un repositorio en GitHub.

ÍNDICE

Introducción

Objetivos

Control de versiones git

Control de versiones GitHub y su integración en VSCode

Conclusiones

OBJETIVOS

Al finalizar esta lección serás capaz de:

- 1 Conocer qué son los Sistemas de Control de Versiones.
- 2 Conocer cuáles son las características principales de los Sistemas de Control de Versiones.
- 3 Saber crear un repositorio en GitHub.
- 4 Clonar y gestionar repositorios de GitHub desde el IDE Visual Studio Code.

Control de versiones Git

Denominamos control de versiones a la gestión de diferentes modificaciones que se han realizado sobre los elementos de algún producto, ya sean códigos de programación o simples documentos de texto.

Una versión hace referencia al estado en el que se encuentra dicho elemento en un momento dado de su desarrollo o modificación.

Como podéis imaginar, un control de versiones puede ser realizado a mano. Una tarea tan simple como la de guardar registros en diferentes directorios o ficheros sobre el trabajo que estemos realizando

Control de versiones Git

El control de versiones se utiliza principalmente en mundo de la informática, más específicamente, controlando las diferentes versiones de código fuente que se va desarrollando a lo largo del proyecto.

Podemos contar con un sistema que nos registra cada cambio, inclusión o borrado de cualquier línea de código de nuestro proyecto, dando lugar a los sistemas de control de código fuente.

Las herramientas de control de versiones también son aplicables a otras áreas de interés como imágenes, documentos, etc.

Control de versiones Git

Todo sistema de control de versiones que se precie debe proporcionar las siguientes características:

- Mecanismo de almacenamiento para los elementos que debe gestionar.
- Posibilidad para realizar cambios sobre los elementos proporcionados.
- Registro histórico de las acciones que se han ido realizando sobre todos y cada uno de los elementos almacenados.
- Generador de informes con los cambios que se han ido introduciendo entre las diferentes versiones.

Control de versiones Git

Terminología

- Repositorio
- Revisión
- Etiquetas (tag)
- Rama (branch)
- Fusión (merge)
- Publicar (commit)
- Conflicto
- Actualización

Control de versiones Git

Arquitecturas de almacenamiento:

- **Distribuidos.** Cada usuario tiene su propio repositorio. Los diferentes repositorios pueden ser intercambiados y mezclados entre ellos. Algunos ejemplos de sistemas de control de versiones distribuidos son Git y Mercurial.
- **Centralizados.** Existe un repositorio centralizado que contiene todo el código, y uno varios usuarios serán los únicos responsables del repositorio. Sacrificando la flexibilidad del sistema de control de versiones, se facilita por contra las tareas administrativas. Algunos ejemplos son CVS o Subversion.

Control de versiones GitHub y su integración en Visual Studio Code

En este apartado aprenderemos a integrar el sistema de control de versiones GitHub con nuestro IDE Visual Studio Code.

Esta integración hace que dispongamos de una forma muy sencilla y amigable de un conjunto de herramientas que nos permitirá desarrollar nuestras implementaciones en Python, así como llevar un control de versiones actualizado sobre nuestro proyecto.

Además, también se presenta la posibilidad de realizar proyectos de forma colaborativa.

Control de versiones GitHub y su integración en Visual Studio Code

Veámoslo mejor con un ejemplo práctico...

CONCLUSIONES

1

Qué son y para qué sirven los sistemas de control de versiones

2

Cuáles son las principales características de los sistemas de control de versiones.

3

Cómo integrar e interactuar GitHub en Visual Studio Code.



MUCHAS GRACIAS POR SU ATENCIÓN



rrueda@grupomainjobs.com



Ramón Rueda Delgado
<https://www.linkedin.com/in/ramon-rueda/>



twitter.com/eiposgrados



facebook.com/eiposgrados



instagram.com/eiposgrados